

reflection_correction.h File Reference

correzione dei riflessi [More...](#)

```
#include <opencv2/imgproc.hpp>
#include <opencv2/opencv.hpp>
```

[Go to the source code of this file.](#)

Functions

void **resize_image** (Mat *input, Mat *output)

Diminuisco width (cols number) dell'immagine input per farla arrivare ad un valore poco ≥ 200 . Della stessa quantità diminuisco anche l'height (rows number). [More...](#)

void **search_reflection** (Mat *input, Mat *mask, int ksize, double c)

Applico un adaptive threshold sul canale blu dell'immagine di input (perché i riflessi sono molto più visibili in questo canale). Sull'output dell'adaptive threshold, applico il dilate, funzione utile per garantire una corretta copertura dei riflessi. In output, alla fine di tale metodo, avrò una maschera binaria contenente i riflessi individuati dell'immagine d'input. [More...](#)

void **inpaint_reflection** (Mat *input, Mat *mask, Mat *output, int iterations)

Applico l'algoritmo di inpainting [TELEA] con maschera "mask", immagine di input "input" e output "output". Tale algoritmo è applicato un numero di volte pari a iterations. [More...](#)

Detailed Description

correzione dei riflessi

Function Documentation

◆ **inpaint_reflection()**

```
void inpaint_reflection ( Mat * input,  
                        Mat * mask,  
                        Mat * output,  
                        int  iterations  
                      )
```

Applico l'algoritmo di inpainting [TELEA] con maschera "mask", immagine di input "input" e output "output". Tale algoritmo è applicato un numero di volte pari a iterations.

Parameters

input immagine di input (ottenuta dall'output di una precedente chiamata di resize_image)
mask immagine di mask (ottenuta da una precedente chiamata di search_reflection)
output immagine di output
iterations quantità di volte in cui viene applicato l'algoritmo di inpainting

◆ resize_image()

```
void resize_image ( Mat * input,  
                  Mat * output  
                )
```

Diminuisco width (cols number) dell'immagine input per farla arrivare ad un valore poco ≥ 200 . Della stessa quantità diminuisco anche l'height (rows number).

Parameters

input immagine di input
output immagine di output

◆ search_reflection()

```
void search_reflection ( Mat *   input,  
                        Mat *   mask,  
                        int      ksize,  
                        double   c  
                      )
```

Applico un adaptive threshold sul canale blu dell'immagine di input (perché i riflessi sono molto più visibili in questo canale). Sull'output dell'adaptive threshold, applico il dilate, funzione utile per garantire una corretta copertura dei riflessi. In output, alla fine di tale metodo, avrò una maschera binaria contenente i riflessi individuati dell'immagine d'input.

Parameters

input immagine di input

mask immagine di output (maschera vuota)

ksize grandezza del kernel (ksize x ksize)

c coefficiente da sottrarre alla media gaussiana pesata